Analysis of Unicorn Startups

Contents

1 Setup

1.1 Import Packages

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
from matplotlib.ticker import FuncFormatter
import seaborn as sns
import re
```

2 Data Preparation

2.1 Load Data

```
pd.set_option('display.max_columns', 50, 'display.width', 200)
```

```
df = pd.read_csv('input/Unicorns_Completed.csv')
```

2.2 Data Cleaning

```
def convert_years_months(s):
    m = re.match(r'(\d+)y?\s?(\d+)m?o?', s)
    return f'{m[1]}y{m[2]}m' if m else s

df['Years to Unicorn'] = df['Years to Unicorn'].apply(convert_years_months)
```

2.3 Prepare data

```
df['Unicorn Date'] = pd.to_datetime(df['Unicorn Date'])
df['Valuation ($B)'] = pd.to_numeric(df['Valuation ($B)'])
```

2.4 Preview data

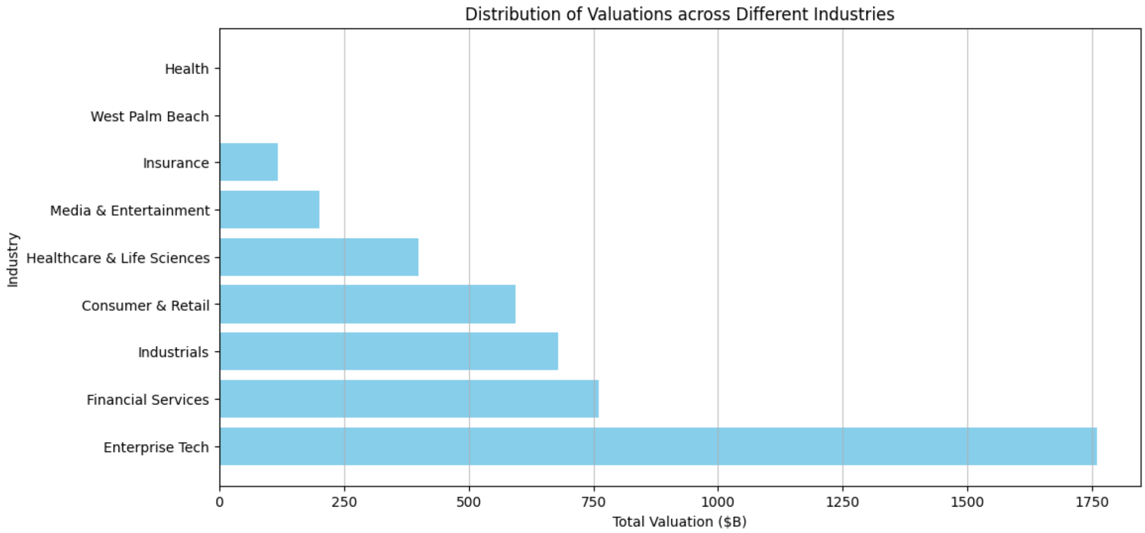
```
df.head()
```

3 Descriptive Analysis

3.1 Distribution of Valuations across Different Industries

```
# Group by industry and sum valuations
industry_valuation_df = df.groupby('Industry')['Valuation
↪  ($B)'].sum().reset_index().sort_values('Valuation ($B)', ascending=False)
industry_valuation_df
```

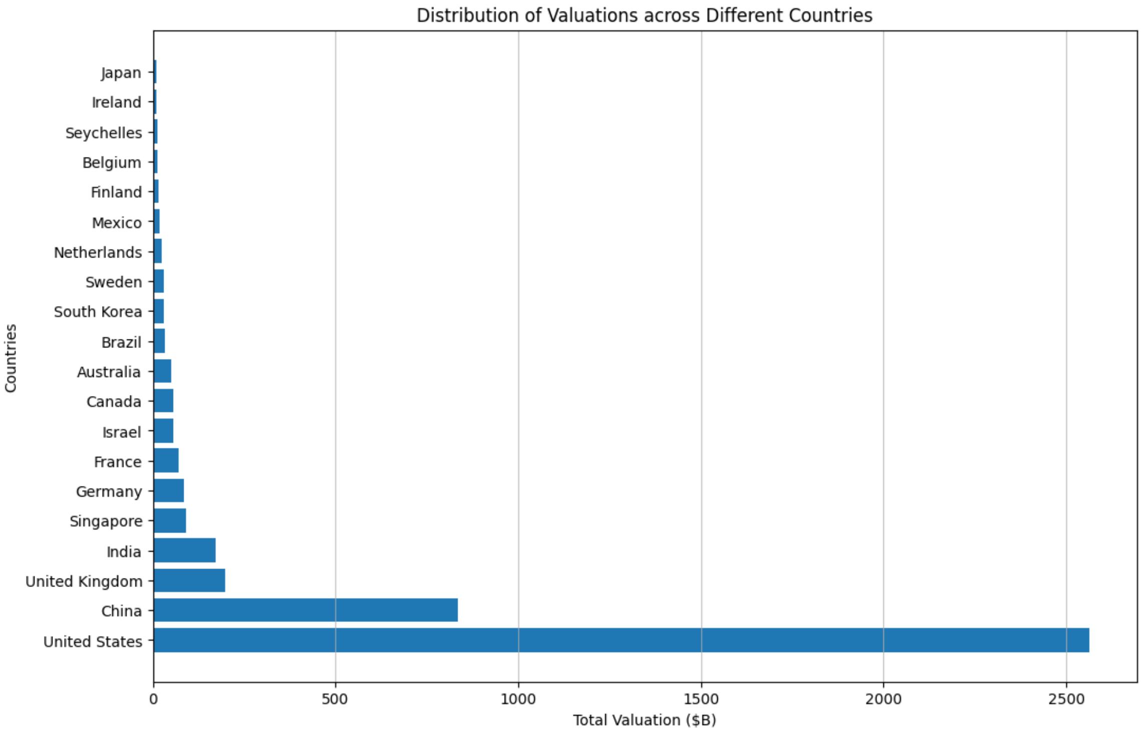
```
plt.figure(figsize=(12, 6))
plt.barh(industry_valuation_df['Industry'], industry_valuation_df['Valuation ($B)'],
↪  color='skyblue')
plt.title('Distribution of Valuations across Different Industries')
plt.xlabel('Total Valuation ($B)')
plt.ylabel('Industry')
plt.grid(axis='x', alpha=0.75)
```



3.2 Distribution of Valuations across Different Countries

```
# Group by Country and sum valuations
country_valuation_df = df.groupby('Country')['Valuation
↪ ($B)'].sum().reset_index().sort_values('Valuation ($B)', ascending=False).head(20)
country_valuation_df
```

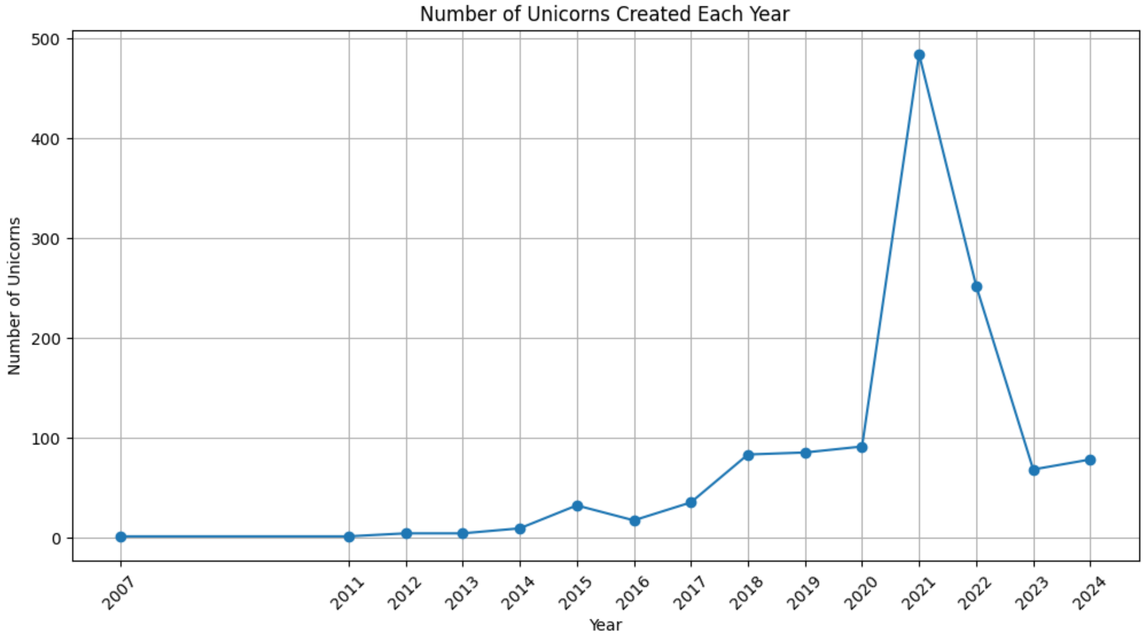
```
plt.figure(figsize=(12, 8))
plt.barh(country_valuation_df['Country'], country_valuation_df['Valuation ($B)'])
plt.title('Distribution of Valuations across Different Countries')
plt.xlabel('Total Valuation ($B)')
plt.ylabel('Countries')
plt.grid(axis='x', alpha=0.75)
```



4 Time-Based Analysis

4.1 Unicorn Growth Over Time

```
unicorn_count = df.groupby(df['Unicorn Date'].dt.year).size()
plt.figure(figsize=(12, 6))
plt.plot(unicorn_count.index, unicorn_count.values, marker='o')
plt.title('Number of Unicorns Created Each Year')
plt.xlabel('Year')
plt.ylabel('Number of Unicorns')
plt.xticks(unicorn_count.index, rotation=45)
plt.grid()
```

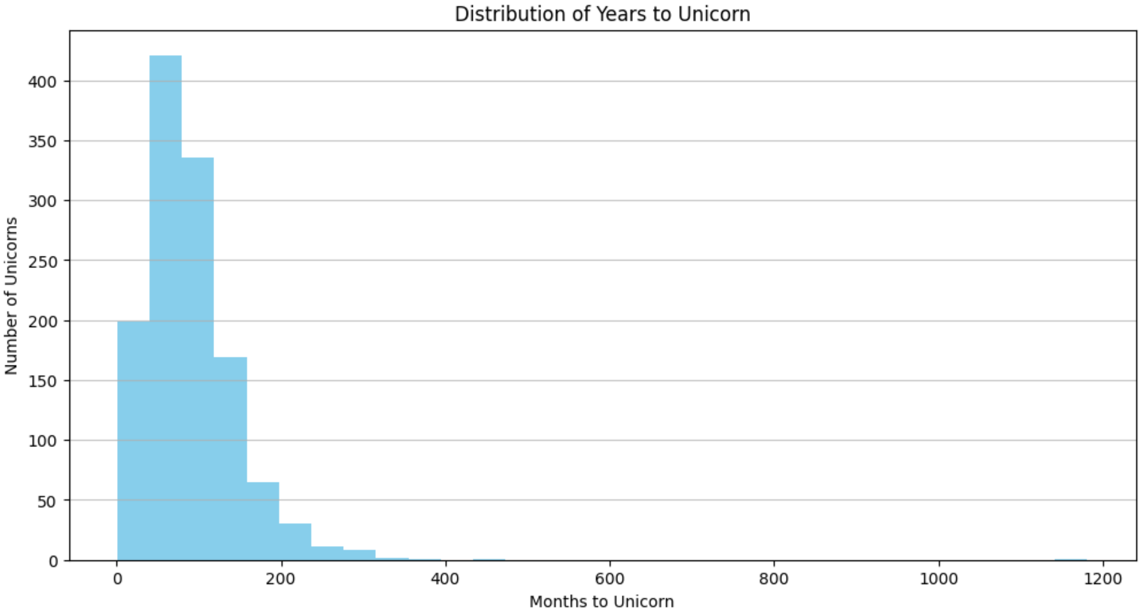


4.2 Years to Unicorn

```python
# Function to convert "Years to Unicorn" into total months
def convert_years_to_months(years_str):
    if 'y' in years_str and 'm' in years_str:
        years, months = years_str.split('y')
        months = months.replace('m', '').strip()
        return int(years.strip()) * 12 + int(months)
    elif 'y' in years_str:
        years = years_str.replace('y', '').strip()
        return int(years) * 12
    elif 'm' in years_str:
        months = years_str.replace('mo', '').replace('m', '').strip()
        return int(months)
    else:
        return None

df['Years to Unicorn (Months)'] = df['Years to Unicorn'].apply(convert_years_to_months)
```

```python
plt.figure(figsize=(12, 6))
plt.hist(df['Years to Unicorn (Months)'].dropna(), bins=30, color='skyblue')
plt.title('Distribution of Years to Unicorn')
plt.xlabel('Months to Unicorn')
plt.ylabel('Number of Unicorns')
plt.grid(axis='y', alpha=0.75)
```


Distribution of Years to Unicorn
Number of Unicorns
0
50
100
150
200
250
300
350
400
0
200
400
600
800
1000
1200
Months to Unicorn